1. Is the Python Standard Library included with PyInputPlus?

Ans1

No, PyInputPlus is a separate package from the Python Standard Library and does not include the standard library modules by default. However, PyInputPlus may import certain modules from the standard library, such as re, datetime, and os, depending on the features used in PyInputPlus.

PyInputPlus is designed to simplify user input validation and processing, and includes its own set of functions and classes for this purpose. It can be installed using pip, and once installed, it can be imported into Python scripts using the import pyinputplus statement.

2. Why is PyInputPlus commonly imported with import pyinputplus as pypi?

Ans2

PyInputPlus is commonly imported with the alias pypi using the syntax import pyinputplus as pypi for brevity and convenience.

Using pypi as an alias saves typing time and reduces the amount of code required to use PyInputPlus in scripts.

3. How do you distinguish between inputInt() and inputFloat()?

Ans3

Both inputInt() and inputFloat() are functions provided by the PyInputPlus package to allow users to input integer and floating-point values, respectively. The main difference between these two functions is the type of input that they accept.

The inputInt() function is used to prompt the user to enter an integer value, and will only accept inputs that can be converted to integers. This means that any non-integer input, such as a floating-point number or a string, will result in a ValueError being raised.

The inputFloat() function, on the other hand, is used to prompt the user to enter a floating-point value, and will accept inputs that can be converted to floats. This means that both integer and floating-point inputs will be accepted, as well as any input that can be converted to a float, such as a string representing a number with decimal places.

4. Using PyInputPlus, how do you ensure that the user enters a whole number between 0 and 99?

Ans4

To ensure that the user enters a whole number between 0 and 99 using PyInputPlus, you can use the inputInt() function with the min, max, and prompt parameters.

import pyinputplus as pyip

# Prompt the user to enter an integer between 0 and 99

user\_input = pyip.inputInt(prompt='Please enter an integer between 0 and 99: ',min=0,max=99)

print(f'You entered: {user\_input}')

5. What is transferred to the keyword arguments allowRegexes and blockRegexes?

Ans5

he keyword arguments allowRegexes and blockRegexes are typically used in the context of web scraping or text processing, to filter or allow certain patterns of text data.

allowRegexes is a keyword argument that is used to pass a list of regular expressions (regexes) that should be allowed. When this argument is used, only the text data that matches one or more of the specified regexes will be included in the output.

blockRegexes, on the other hand, is a keyword argument that is used to pass a list of regexes that should be blocked. When this argument is used, any text data that matches one or more of the specified regexes will be excluded from the output.

6. If a blank input is entered three times, what does inputStr(limit=3) do?

Ans6

If a blank input is entered three times when using the inputStr(limit=3) function, it will raise a ValueError with the message "Exceeded maximum number of tries".

The inputStr() function with the limit parameter is used to prompt the user to input a string, with the number of tries limited by the limit parameter. In this case, limit is set to 3, so the user will have up to three tries to input a non-blank string.

7. If blank input is entered three times, what does inputStr(limit=3, default=&#39;hello&#39;) do?

Ans7

If a blank input is entered three times when using the inputStr(limit=3, default='hello') function, it will return the default value "hello".

The inputStr() function with the limit and default parameters is used to prompt the user to input a string, with the number of tries limited by the limit parameter. In addition, the default parameter specifies a default value to return if the user enters a blank string and exceeds the maximum number of tries.

In this case, limit is set to 3 and default is set to "hello". If the user enters a non-blank string within the allowed number of tries, the function will return the string that was inputted. However, if the user enters a blank string the function will prompt the user again for input, and this will repeat until either the user inputs a non-blank string or the maximum number of tries is reached.

The inputStr(limit=3, default='hello') function will always return a non-blank string, either the one entered by the user or the default value "hello" if the user exceeds the maximum number of tries.